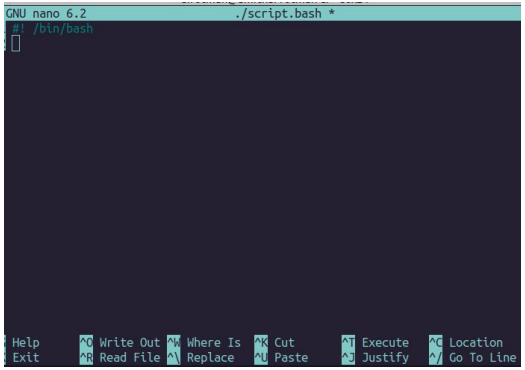
BASH Basics

BASH Shell Basics for ROS Users

What's a Shell?

What's a Shell?



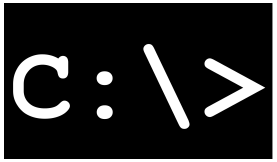
- A Shell is a command interpreter.
- It's the first program loaded into memory as soon as the user logs on.
- The Shell includes a few basic commands, providing access to the rest of the OS.
- Simply put: Without it, the user cannot use a GNU/Linux system...

Things to consider when using ROS

- You'll have to use the Terminal (CLI) - ***alot***...
- You'll need to edit configuration files.
- You'll have to learn how to manipulate environment variables.
- You'll use the keyboard more than the mouse.
- Using keyboard shortcuts saves time and prevent mistakes.

Different Ways to access the BASH Shell

On Microsoft Windows

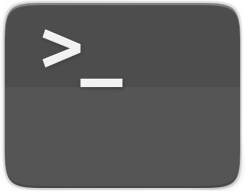


There are a few ways to access the BASH Shell using Microsoft Windows:

- By installing WSL 2.0.
- By connecting to a GNU/Linux machine remotely, via Secure Shell (SSH), using the built in SSH program (accessible via the command prompt (CMD)) or PUTTY*.

*PUTTY is quite old and hasn't been updated for a while, thus, we recommend using the built in SSH program.

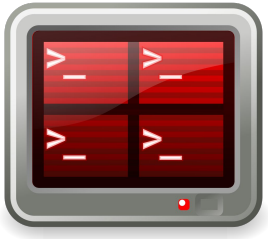
On GNU/Linux



The BASH Shell is part of most modern GNU/Linux distributions, so, all you need to do to access a BASH Shell is open the default Terminal emulator by pressing the following keyboard shortcut:

CTRL+ALT+T

The Terminator Emulator



Working with ROS requires you to use multiple Terminal windows, each running a different command. Using a terminal Multiplexer emulator makes it much easier, by allowing you to run a few terminal instances one split window. That is why we recommend you to install Terminator, by entering the following in a terminal:

```
$ sudo apt install terminator
```

Connecting to a remote BASH Shell

To connect to a remote GNU/Linux machine, open a CMD window and enter the following:

```
ssh <user>@<remote host IP>
```

When connecting to a remote Technion machine

- Make sure you are connected to the TechSec WiFi network / Technion LAN port or the Technion VPN
- Make sure you remember you Technion ID* and password.

* *For example: <user>@campus.technion.ac.il*

A note about the BASH Prompt

There are two possible BASH Prompts, each indicating a different user access level to the system:

`$`	`#`
Regular User	Root User

Differences between Root and Regular users

Root User

- ✓ Has full access to the Operating System
- ✓ Can manage users.
- ✓ Can Manage software.

Regular User

- ✓ Has access to only his/her own files (in the /home directory).
- ✓ Can't manage system resources
- ✓ Can't manage software.

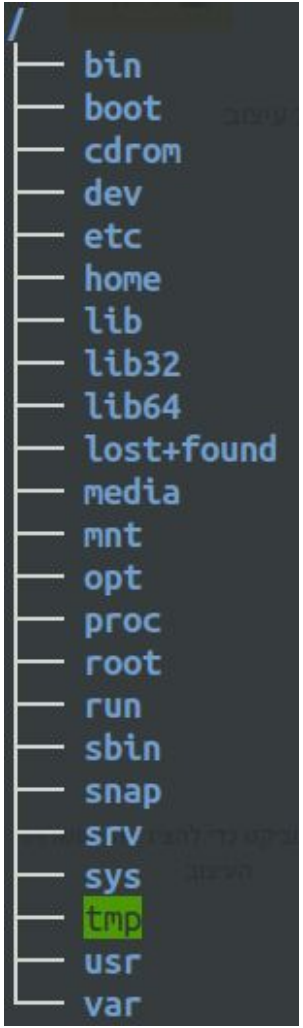
Some Basic BASH Commands

Finding your way around: PWD

Finding your way around: pwd

PWD: **P**rint **W**orking **D**irectory

The GNU/Linux File System is set up like an upside down tree, with the root (/) directory on top, and the rest of the files branching out from it.



Finding your way around: `pwd` (contd.)

We can sometimes find identical files in different locations of the file system, thus, it is good practice to make sure we are in the right place.

We can do that, simply, by entering `pwd` at the command prompt.

Finding your way around: pwd (contd.)

Enter the following at the command prompt:

```
$ pwd
```

Finding your way around: `pwd` (contd.)

As you can see, the command prints your working directory, relative to the root (/) directory, sort of like turning the light on for a brief moment in a dark room ;-)

Creating New Directories: MKDIR

Creating New Directories: MKDIR

Creating a new directory is like placing a new cabinet in our room. Once it's there, we can use it to put objects (=files) in it.

To do this we use the `mkdir` command.

Creating New Directories: MKDIR (contd.)

To create our first new directory, enter the following at the command prompt:

```
$ mkdir./bin
```

📝 :*The* `./` *before the directory name means "Current Directory".*

Moving Around: CD

Moving Around: CD

Now we can put what we've learned so far into practice, and start moving around the file system!

The **C**hange **D**irectory command allows us to move to another, completely different, directory (or branch) of the tree.

We can even jump between branches! ;-)

Moving Around: CD (contd.)

Enter the following at the command prompt:

```
$ cd
```

Moving Around: CD (contd.)

Next, enter the following at the command prompt:

```
$ pwd
```

Moving Around: CD (contd.)

Next, enter the following at the command prompt:

```
$ cd ./bin
```

Moving Around: CD (contd.)

Next, enter the following at the command prompt:

```
$ pwd
```

Moving Around: CD (contd.)

Next, enter the following at the command prompt:

```
$ cd /bin
```

Moving Around: CD (contd.)

Next, enter the following at the command prompt:

```
$ pwd
```

Moving Around

What happened?

Moving Around: CD`(contd.)`

OK, so we've made a mistake!

Let's fix it...

Moving Around: CD (contd.)

Next, enter the following at the command prompt:

```
$ cd -
```

📝 :*That is, cd, space, and a dash (-)*

Moving Around: CD (contd.)

Next, enter the following at the command prompt:

```
$ pwd
```

Moving Around: CD(contd.)

😎 Cool, right?

Moving Around: CD(contd.)

OK, say we got tired and we want to go home...

Moving Around: CD(contd.)

So, next, enter the following at the command prompt:

```
$ cd
```

Moving Around: CD (contd.)

Next, enter the following at the command prompt:

```
$ pwd
```

Searching History: CTRL+R

Searching History: CTRL+R

- ★ The BASH Shell keeps track of all the commands we enter at the command prompt.
- ★ To search the command history, we use the `CTRL+R` key combination.

Searching History: CTRL+R (contd.)

At the terminal window, press CTRL+R, and start typing:

```
$ pw
```

Searching History: CTRL+R(contd.)

To execute the command, just press the `Enter` key.

Echo in the dark...

Echo in the dark

- ★ The shell uses environment variables to track dynamic changes.
- ★ To find out the value of a variable we use the `echo` command.

Echo in the dark(contd.)

Enter the following at the command prompt:

```
$ echo $USER
```

: *The GNU/Linux command line is case sensitive!*

Creating a File: Touch

Creating a File: Touch

If we want to create a file that doesn't exist, we can use the `touch` command.

Creating a File: Touch(contd.)

Enter the following at the command prompt:

```
$ touch ~/bin/test
```

📝: `~` *stands for the home directory (`/home/$USER`)*

Listing Files: LS

Listing Files: LS

To list the files at a given directory, we use the `ls` command. There are a lot of ways to list files, for now we will try the two most common ways.

Listing Files: LS(contd.)

Enter the following at the command prompt:

```
$ ls ./
```

Listing Files: LS(contd.)

Next, enter the following at the command prompt:

```
$ ls -la ./
```

The Nano Text Editor

The Nano Text Editor

Another way to create and edit text files at the console is the nano text editor. It is the most user friendly way to create and edit files at the command prompt.

The Nano Text Editor (contd.)

I trust you'll find your way in nano easy enough, but here are a few hints...

The Nano Text Editor `(contd.)`

- ★ All the command are presented at the bottom of the screen.
- ★ The carrot (^) refers to the `CTRL` key.
- ★ You can access a short help introduction by pressing `CTRL+G`.
- ★ The most important tip: READ.

THANK YOU!

I CAN BE REACHED AT:

The Cognitive Robotics Laboratory,Cooper BLDG, Room #300
SUN-THURS | 10:30-18:30

a.rotman@technion.ac.il

@AmichaiRotman

Thank You!